Autonomous Vocal and Backing Track Mixing

Kelian (Mike) Li

Georgia Tech Center for Music Technology College of Design

Introduction

The field of automatic mixing is growing over the years, and the approaches are moving from rule-based expert systems to deep learning. While most research focuses on automatic mixing on multi-tracks, the area of vocal and backing track mixing is largely ignored. A rule-based baseline implementation closes the gap by performing autonomous level balance, compression, equalization, and reverberation. A trial experiment of level balance by deep learning shows that the model is not yet learning.

Rule-based Method

Level Balance

- The targeted related loudness of the vocal and the backing track is **-0.5 dB**, and the resulted relative loudness of the vocal track and the full mix falls near **-3 dB**.

Compression

- An iterative process optimize the vocal loudness range to **15.7 dB ± 1 dB** by modifying the compression threshold and compression ratio.
- The compression threshold and the compression ratio are initialized to 0 dB and 3:1. For each iteration, the compression threshold is reduced by 3 dB, and the compression ratio is increased by 0.1. The process ends if the loudness range is below 16.7 dB, or the current compression threshold is less than -35 dB.

Equalization

- The automatic equalization is achieved by the frequency masking reduction method proposed by Hafezi and Reiss [1].
  - (1) the frequency band is the essential band of the maskee signal;
  - (2) the frequency band is the nonessential band of the masker signal;
  - (3) the masker signal masks the maskee signal in this frequency band.
- When the vocal is the masker, the peak filter applies a negative gain (cut). When the vocal is the maskee, the peak filter applies a positive gain (boost).

Algorithm 1 frequency unmasking

apply noise gate to lead vocal vox and the backing track bac

VOX_{band}, BAC_{band} = frequencyBandRMS(vox, bac)
$DIFF_{band}$ = $VOX_{band} - BAC_{band}$
$DIFF_{band}$ −= mean($DIFF_{band}$)

vox^*, acc^* = KFilter(vox, bac)
VOX^*_{band}, BAC^*_{band} = frequencyBandRMS(vox^*, bac^*)
VOX_{idx} = argsort(VOX^*_{band})
BAC_{idx} = argsort(BAC^*_{band})

if idx in top 4 VOX_{idx} **then**
  if idx not in top 3 BAC_{idx} **then**
    if $DIFF_{band}[idx] < 0$ **then**
      $gain = DIFF_{band}[idx]$
      $freq = band[idx]$
      apply peak filter on vox
    end if
  end if
end if

if idx in top 3 ACC_{idx} **then**
  if idx not in top 4 VOX_{idx} **then**
    if $DIFF_{band}[idx] > 0$ **then**
      $gain = DIFF_{band}[idx]$
      $freq = band[idx]$
      apply peak filter on vox
    end if
  end if
end if

[1] S. Hafezi and J. D. Reiss, "Autonomous multitrack equalization based on masking reduction," *Journal of the Audio Engineering Society*, vol. 63, no. 5, 2015.
[2] D. Moffat and M. Sandler, "An automated approach to the application of reverberation," in *Audio Engineering Society Convention*, 2019.
[3] B. De Man, K. McNally, and J. D. Reiss, "Perceptual evaluation and analysis of reverberation in multitrack music production," *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, 2017.

Reverberation

- Moffat and M. Sandler [2] present a set of mixing rules that maps audio features to reverb parameters. Our rule-based reverberation linearly maps the tempo to the reverb time.

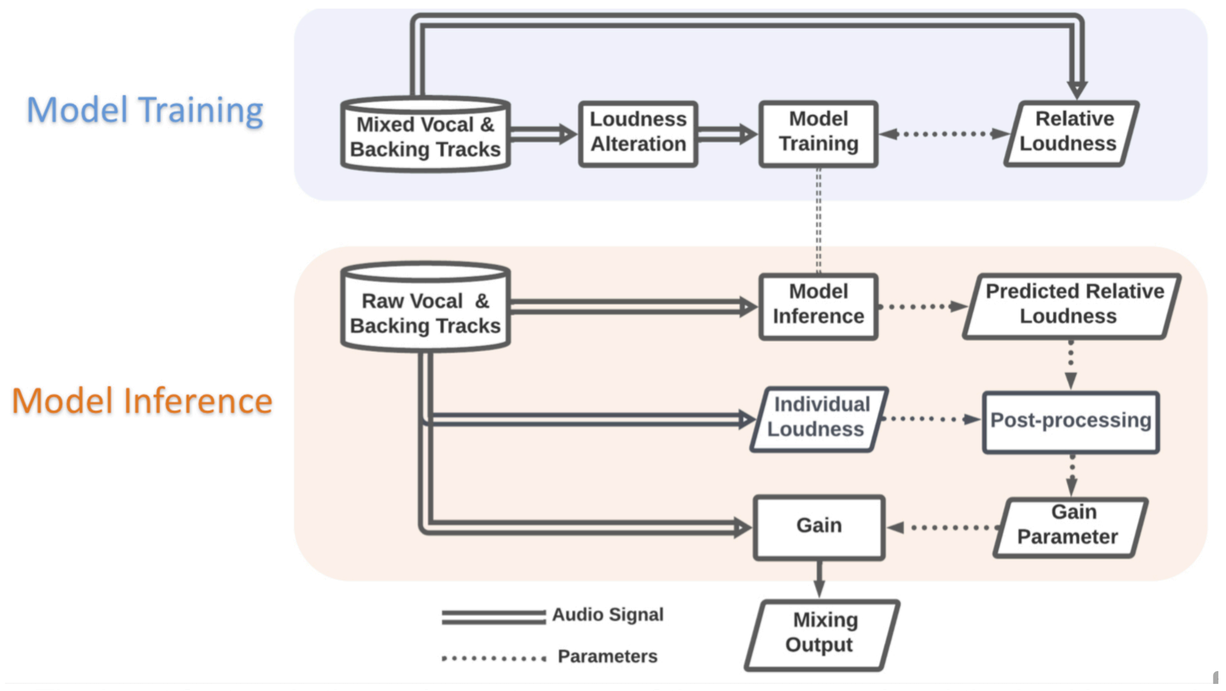
$$RT = -\frac{1}{45}T + \frac{40}{9}$$

RT is the reverb time, and T is the tempo.

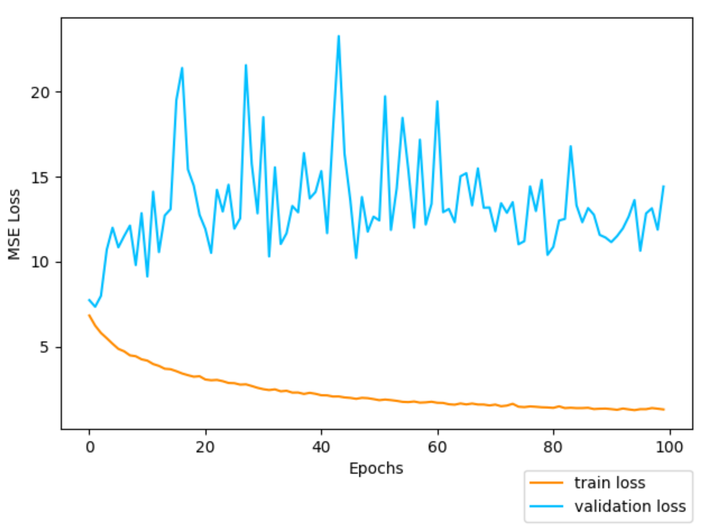
- De Man et al. [3] show that the reverb loudness are typically -14 dB lower than the full mix. In our implementation, the dry/wet ratio is set to 60%.

Data-driven Level Balance

The data-driven approach aims to train a regression model which is able to predict the ideal mixing parameters based on the raw audio features of the input signal.



The input feature is the mel-spectrogram of the mono vocal and the backing track. each block represents around 1.51 seconds. The size of the input block is (2, 128, 64). The ground truth target is the relative loudness of the two tracks over the full song. The training and validation data is the professional produced vocal tracks and backing tracks from MUSDB18HQ dataset



Conv2d (input channels = 2, output channels = 8)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 8, output channels = 16)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 16, output channels = 32)
BatchNorm
ReLU
MaxPool2d
Conv2d(input channels = 32, output channels = 64)
BatchNorm
ReLU
MaxPool2d
Dropout (p = 0.3)
MLP (input features = 768, output features = 1)

The absolute error is 1.14 dB and 3.80 dB for training and validation after 100 epochs. The validation loss indicates that the model is not learning at all.

The failure of the trial experiment may be due to the following reasons:

(1) the lack of generalization ability of this model;
(2) not enough training data;
(3) improper input features;
(4) improper ground truth representation;
(5) the challenge of the task itself;
(6) code errors.

CONTACT

Kelian (Mike) Li
Music Informatics Group
Center for Music Technology
Georgia Tech
kli421@gatech.edu

https://github.com/likelian/Lab_spring2022